

Creating and Modifying an Isoline Graphics Method

Â

[Contents](#) [Previous](#) [Next](#)

Goal: Guide you through creating and setting isoline graphics method attributes.

Before running the tutorial below, type "*python*" or "*cdat*" at the command line.Â You will see the python prompt appear (i.e., ">>>"). You can now enter the command lines below.

You can [view](#)Â or [download](#)Â the full source code. To run the source code at the command line, type: "*python isoline_file.py*".

```
# Import the modules needed for the tutorial
# cdms - Climate Data Management system accesses gridded data.
# vcs - Visualization and control System 1D and 2D plotting routines.
# cdutil - Climate utilitizes that contains miscellaneous routines for
#           manipulating variables.
# time - This module provides various functions to mainpulate time values.
# os - Operation System routines for Mac, DOS, NT, or Posix depending on
#       the system you're on.
# sys - This module provides access to some objects used or maintained by
#       the interpreter and to functions that interact strongly with the interpreter.
import vcs, cdms, cdutil, time, os, sys

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/clt.nc')
cdmsfile = cdms.open( filepath )

# Extract a 3 dimensional data set and get a subset of the time dimension
data = cdmsfile('clt', longitude=(-180, 180), latitude = (-90., 90.))

# Initial VCS:
v = vcs.init()

# Show the list of persistent isoline graphics methods.
v.show('isoline')

*****Isoline Names List*****
( 1):          ASD      P_and_height      default
( 4):          map      polar            quick
*****End Isoline Names List*****
```

Get a isoline graphics method object and plot:

```
# Assign the variable "df_asd" to the persistent 'ASD' isoline graphics methods.
df_asd = v.getisoline( 'ASD' )

# Plot the data using the above isoline graphics method.
v.plot( data, df_asd )
```

List the 'ASD' isoline graphics method attributes by issuing the following command:

```
# List the 'ASD' isoline graphics methods attributes.  
df_asd.list()  
  
-----Isoline (Gi) member (attribute) listings -----  
Canvas Mode = 1  
graphics method = Gi  
name = ASD  
projection = linear  
xticlabels1 = *  
xticlabels2 = *  
xmtics1 =  
xmtics2 =  
yticlabels1 = *  
yticlabels2 = *  
ymtcs1 =  
ymtcs2 =  
datawc_x1 = 1.00000002004e+20  
datawc_y1 = 1.00000002004e+20  
datawc_x2 = 1.00000002004e+20  
datawc_y2 = 1.00000002004e+20  
datawc_timeunits = days since 2000  
datawc_calendar = 135441  
xaxisconvert = linear  
yaxisconvert = linear  
label = n  
line = ['solid']  
linecolors = [241]  
linewidths = [1.0]  
text = None  
textcolors = None  
level = [[0.0, 1.0000000200408773e+20] ]
```

Change 'ASD' isoline graphics methods attributes by entering the appropriate command lines:

```
# change the isofill levels and line type and color  
df_asd.levels = ([20,0], [30,0], [40,0], [50,0], [60,0])  
df_asd.levels = ([0,20], [20,40], [50,60])  
df_asd.levels = ([20,0], [30,0], [40,0], [50,0], [60,0])  
df_asd.levels = (30,50,70)  
df_asd.line=[0, 2, 0]  
df_asd.linecolors=(16,100,200)  
  
  
# view the level labels  
df_asd.label='y'  
  
  
# set the label font and text color  
df_asd.text=(1, 5, 9)  
df_asd.textcolors=(16,100,200)  
  
  
# Create a persistent isoline graphics methods from an existing isoline graphics method.  
cf_new = v.createisoline( 'new', 'ASD' ) # create new from ASD
```

```
cf_new2 = v.createisoline( 'new2', 'quick' )# create new2 from quick
cf_new.list()                                # list its attributes
v.show('isoline')                            # show isoline list with new and new2
v.removeobject( cf_new )                      # remove new from isoline list
v.show('isoline')                            # show isoline list without new
```

Â

[Contents](#) [Previous](#) [Next](#)